课后作业二：概率分类法

任务：使用贝叶斯估计或 MLE（最大似然估计），来预测鸢尾花数据集中花的种类。

数据集：鸢尾花数据集是统计学和机器学习中用于分类的经典数据集。该数据集包含了三种不同的鸢尾花：Setosa、Versicolor 和 Virginica，每种各 50 个样本。每个样本有四个属性：萼片长度、萼片宽度、花瓣长度和花瓣宽度，所有的测量单位都是厘米。数据集根据 4:1 的比例划分为训练集和测试集。概率分类法是一种基于概率理论的方法，适合处理此类分类问题。

原理：

贝叶斯定理公式：

$$P(C_k|x) = \frac{P(x|C_k) \cdot P(C_k)}{P(x)}$$

其中：

$P(C_k|x)$ 是给定特征 x 属于类别 C_k 的后验概率。

$P(x|C_k)$ 是给定类别 C_k 的情况下特征 x 的似然。

$P(C_k)$ 是类别 C_k 的先验概率。

$P(x)$ 是特征 x 的边际概率。

实现：

要实现通过贝叶斯分类进行预测，需要完成以下几步

- 计算每个类别的先验概率。
- 对每个类别和每个特征，计算其均值和标准差。
- 使用高斯分布计算给定特征的似然。
- 使用贝叶斯定理计算后验概率。
- 选择具有最高后验概率的类别作为预测类别。

1.导入必要的库

```
from sklearn.model_selection import train_test_split     #机器学习中非常重要的库，包括一些分类、回归、聚类、降维、模型选择和预处理

from collections import defaultdict
from math import sqrt, pi, exp
import pandas as pd     #用来分析结构化数据
import numpy as np      #提供高性能的矩阵运算
import csv    #读写文件的库
```

2.导入训练数据并提取特征值和目标值

"sepal length (cm)", "sepal width (cm)", "petal length (cm)", "petal width (cm)"为特征值，"species"为目标值

```
iris_data = pd.read_csv(r'D:\dataenclorse\second\iris_train.csv')
X = iris_data[["sepal length (cm)", "sepal width (cm)", "petal length (cm)", "petal width (cm)"]].values
y = iris_data["species"].values
```

3.划分数据

```
# 构造训练数据和测试数据
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)
```

4.构造贝叶斯分类模型并训练该模型

（1）定义计算先验概率的函数

```
# 计算每个类别的先验概率
def calculate_prior(y):
    class_counts = defaultdict(int)
    for label in y:
        class_counts[label] += 1
    total_count = len(y)
    priors = {label: count / total_count for label, count in class_counts.items()}
    return priors
```

（2）手动实现计算特征值和标准差的函数

```
# 计算每个类别和每个特征的均值和标准差
def calculate_mean_std(X, y):
    separated = defaultdict(list)
    for i in range(len(y)):
        separated[y[i]].append(X[i])
    summary = {}
    for label, instances in separated.items():
        summary[label] = [(np.mean(attribute), np.std(attribute)) for attribute in zip(*instances)]
    return summary
```

（3）实现高斯分布的概率密度函数

```
def gaussian_probability(x, mean, std):
```

```
    exponent = exp(-((x - mean) ** 2 / (2 * std ** 2)))
    return (1 / (sqrt(2 * pi) * std)) * exponent
```

（4）实现计算特征的似然的函数

```
# 计算给定特征的似然
def calculate_likelihood(summary, x):
    likelihoods = {}
    for label, stats in summary.items():
        likelihood = 1
        for i in range(len(stats)):
            mean, std = stats[i]
            likelihood *= gaussian_probability(x[i], mean, std)
        likelihoods[label] = likelihood
    return likelihoods
```

（5）使用贝叶斯定理计算后验概率

```
def calculate_posterior(priors, likelihoods):
    posteriors = {}
    for label in priors:
        posteriors[label] = priors[label] * likelihoods[label]
    total_posterior = sum(posteriors.values())
    for label in posteriors:
        posteriors[label] /= total_posterior
    return posteriors
```

（6）手动实现的朴素贝叶斯分类器预测函数

```
def naive_bayes_predict(X_train, y_train, X_new):
    priors = calculate_prior(y_train)
    summary = calculate_mean_std(X_train, y_train)
    predictions = []
    for x in X_new:
        likelihoods = calculate_likelihood(summary, x)
        posteriors = calculate_posterior(priors, likelihoods)
        best_label = max(posteriors, key=posteriors.get)
        predictions.append(best_label)
    return predictions
```

5.加载 iris_test.csv 的数据并对 iris_test.csv 进行预测

```
# 读取待预测的新数据点
iris_test_data = pd.read_csv(r'D:\dataenclorse\second\iris_test.csv')
X_new = iris_test_data[["sepal length (cm)", "sepal width (cm)", "petal length (cm)", "petal width
(cm)"]].values

# 预测新数据点的类别
predictions = naive_bayes_predict(X_train, y_train, X_new)
print("预测的目标类别是：{}".format(predictions))
```

6.预测结果与加载的数据一起保存到 test.csv 文件中

getdata 和 getdata2 函数与作业一中一模一样，仅仅有读取文件的内容不同

```
file_path = r'D:\dataenclorse\second\test_manual_bayes.csv'
with open(file_path, 'w', encoding='utf-8', newline='') as f:
    fieldnames = ['sepal length(cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)',
'class']
    f_csv = csv.DictWriter(f, fieldnames=fieldnames)
    f_csv.writeheader()
    for i in range(len(predictions)):
        f_csv.writerow({
            'sepal length(cm)': X_new[i][0],
            'sepal width (cm)': X_new[i][1],
            'petal length (cm)': X_new[i][2],
            'petal width (cm)': X_new[i][3],
            'class': predictions[i]
        })
print(f"预测结果已保存到 {file_path}")
```

7. 结果如下：

```
(pytorch) D:\dataenclorse>python -u "d:\dataenclorse\second\second.py"
预测的目标类别是：[1, 0, 2, 2, 1, 1, 1, 2, 1, 1, 1, 0, 0, 0, 0, 1, 2, 1, 2, 2, 1, 2, 0, 2, 2, 2, 2, 2, 0, 1]
预测结果已保存到 D:\dataenclorse\second\test_manual_bayes.csv
```

	A	B	C	D	E
	sepal len	sepal wid	petal len	petal wid	class
	5.8162431	2.550183	5.0114162	0.4624863	1
	5.8100076	4.3979506	1.3004943	0.30699	0
	8.1375468	3.0778532	7.550411	2.1863261	2
	5.2453112	2.943514	4.0458649	2.075542	2
	6.9100196	3.0875745	4.8606287	0.9719949	1
	5.1617493	3.9913823	2.1057935	0.769319	1
	5.5034111	3.1224913	4.116424	1.8540749	1
	6.8831622	3.0830352	5.6621994	2.1921701	2
	6.4285034	1.9282514	4.8793979	1.5197273	1
	6.0816364	2.4118773	3.4406134	1.3768565	1
	6.0735042	2.9357557	4.547916	1.6788906	1
	4.1280984	3.4539907	1.5299596	0.2326497	0
	5.9342591	3.9596878	0.7358813	0.0470084	0
	5.1763616	3.204539	1.2029467	-1.252725	0
	5.3239954	3.8175243	1.7286487	0.5836608	0
	6.4081433	3.8030474	4.0725766	0.6423383	1
	7.3972944	2.6968336	5.9593665	2.1526159	2
	5.006388	1.6894347	2.9609973	1.4042295	1
	6.1919004	2.1313759	4.7942792	1.8795405	2
	6.1093193	2.3811147	5.5773693	2.3878825	2
	4.3330253	3.303213	2.791308	0.2800472	1
	6.2076257	3.1410137	5.1666669	1.4912807	2
	4.9362235	3.6321525	1.8658611	-0.013043	0
	6.6849524	2.8475112	4.7418931	1.7916818	2
	7.5248511	3.7826922	6.4412034	2.5403967	2
	6.5936434	2.7914832	5.1344312	1.8944077	2
	6.6942635	2.1277791	5.7867405	1.9922763	2
	7.1895987	2.2001647	6.351144	2.1673878	2
	4.7193036	2.5926936	1.3643252	0.1730659	0
	4.5589682	3.2530239	1.7432178	1.0025551	1